RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation

Michał Bień[1], **Michał Gilski**[1], **Martyna Maciejewska**[1], **Wojciech Taisner**[1], **Dawid Wiśniewski**[1], **Agnieszka Ławrynowicz**[1,2]
[1]Faculty of Computing and Telecommunications
[2]Center for Artificial Intelligence and Machine Learning (CAMIL)
Poznan University of Technology, Poland

Abstract

Semi-structured text generation is a non-trivial problem. Although last years have brought lots of improvements in natural language generation, thanks to the development of neural models trained on large scale datasets, these approaches still struggle with producing structured, context- and commonsense-aware texts. Moreover, it is not clear how to evaluate the quality of generated texts. To address these problems, we introduce *RecipeNLG* – a novel dataset of cooking recipes. We discuss the data collection process and the relation between the semi-structured texts and cooking recipes. We use the dataset to approach the problem of generating recipes. Finally, we make use of multiple metrics to evaluate the generated recipes.

1 Introduction

A cooking recipe is a very specific category of text, that facilitates sharing culinary ideas between people and provides algorithms for food preparation. Although the recipes follow a set of informal rules which make the cooking experience understandable and reproducible (Fisher, 1969), there are no strict rules on how this text should be structured. This makes it hard to estimate the recipe quality using any objective measures.

Recently, we have noticed a major growth of interest in using cooking recipes datasets for performing deep learning experiments. In particular, there is a number of interesting endeavors utilizing computer vision for finding (Salvador et al., 2017) or even generating (Salvador et al., 2019) cooking recipes matching the input food image. One of the results was the publication of the *Recipe1M+* (Salvador et al., 2017) (Marin et al., 2019) dataset containing both recipes and images. This dataset, which was the largest publicly available recipes dataset at the time, boosted research in this area.

However, while the demand is still emerging, there is currently no large scale cooking dataset tailored specifically for NLP tasks. The existing resources are either not sufficiently big to make efficient use of state of the art language models, or were created with computer vision in mind. In our work, we propose a novel dataset that builds on that previous work and resources. We hope that this resource, which is currently the largest cooking recipes dataset publicly available, may further empower research in the area.

This work is composed of three parts. In Section 3 we outline the problem of imitating cooking recipes and their structure. We show the limitations that caused us to recognize the existing resources as insufficient for generating complete cooking recipes. In Section 4, we introduce a novel recipes dataset built for semi-structured (Buneman, 1997) text generation, which contains over 2 million recipes. We present detailed information about the process of data gathering, deduplication, and cleansing. Finally, in Section 5 we present the implementation details and results of our experiment. We make use of a Named Entity Recognizer (NER) to extract food entities from the dataset and provide them as an input for the recipe generator, using special control tokens. This data is used to fine-tune a GPT-2 (Radford et al., 2019) language model which generates new recipes based on the given list of food entities. We use a number of evaluation methods to compare the generated output to the real recipes using the same set of food entities.

In summary, our work introduces *RecipeNLG* [1] - the novel dataset of cooking recipes, along with the language generation task based on this dataset.

2 Related work

Dissemination of artificial neural network architectures like GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019) or LSTM (Hochreiter and Schmidhuber, 1997), (Merity et al., 2017) allowed to ad-

[1] recipenlg.cs.put.poznan.pl

Proceedings of The 13th International Conference on Natural Language Generation, pages 22–28,
Dublin, Ireland, 15-18 December, 2020.

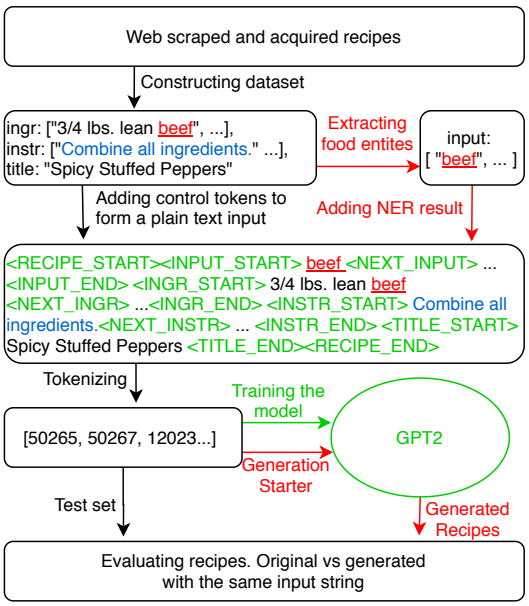

Figure 1: Concept schema of the semi-structured text evaluation pipeline.

vance the field of text generation. Recent developments in neural network architectures (Krizhevsky et al., 2012), (Liang and Hu, 2015) have enabled images to text conversion and vice versa. Publishing *Recipe1M+* dataset (Salvador et al., 2017) made it reasonable to utilize deep neural networks and initiated a series of new publications.

(Marin et al., 2019) combined the *Recipe1M+* dataset with 13 million food images to generate joint embeddings of recipes and images. Their goal was to maximize the coherence of the generated text with its corresponding image. (Bossard et al., 2014) recognized and classified food images into 101 food categories, utilizing a dataset consisting of approximately 100K images. (Salvador et al., 2019) used the *Recipe1M+* to generate simplified recipes lacking ingredient quantities and units. They evaluated their model using a perplexity score as well as the adequacy between the generated text and the image.

A number of efforts are underway to utilize neural language models on recipes datasets. (Parvez et al., 2018) used a dataset of 100K recipes to build an LSTM-based discriminative language model for the task of named entity recognition. They utilized a cooking recipes dataset for evaluation. (Yang et al., 2017) used a dataset with 31K recipes to propose reference-aware language models to generate instructions based on the ingredients provided. (Kiddon et al., 2016) presented a recurrent neural network that models global coherence. It was used to generate individual instructions based on the title and the list of ingredients. They utilized a dataset with 150K cooking recipes for model evaluation. (Yagcioglu et al., 2018) published a dataset consisting of approximately 20K recipes to generate question-answer pairs. (Chandu et al., 2019) built a custom dataset of food images and made use of the text to image approach to perform a storyboarding task for each recipe step. (Luis Herranz and Jiang, 2018) surveyed different approaches to the problem of food recognition and recipe analysis. They published a list of datasets, reported in the literature and their characteristics.

(Majumder et al., 2019) proposed the task of personalized recipe generation, and have shared a dataset of 180K recipes and 700K user interactions (reviews). The authors used an encoder-decoder framework to generate recipes and conducted an evaluation using text metrics. They encoded three embedding layers: title, ingredient, and caloric-level using BERT then decoded recipes steps using a two-layered GRU. (Lee et al., 2020) have recently presented demo paper of their system for the automatic generation of cooking recipes utilizing the *Recipe1M+* dataset and a language model. The evaluation of the model was based on translation metrics. They focused on two separate tasks: ingredients, and instructions generation. On the contrary, we use prepared food entities (see Section 5.1) to generate complete recipes, which allows pairwise comparison of the original and generated recipe composed of the same set of ingredients.

We also propose a new task of generating full recipes with quantities and units. We publish a carefully prepared *RecipeNLG* dataset containing both recipes and tagged food entities, to ease the process of generating and evaluating recipes.

3 Recipes as datasets

Cooking recipes have a specific format which consists of: a title, a list of ingredients with given amounts, and the instructions in a step by step format. The shortest part of the recipe, the title, should accurately name it and summarize its content.

The ingredients list has to contain entities consisting of the quantity, unit name, and ingredient name. The quantities of all ingredients have to be in line with the number of servings the recipe is made for. The unit name has to be in relation to the quantity. It must be appropriate to the ingredient form (liquid, dry countable, dry uncountable).

Finally, all the units in the recipe are expected to follow a single unit system - imperial or metric.

The instructions section needs to accurately present the order of steps. The actions performed on every ingredient have to be taken into account in the following recipe steps, which should reflect the state of the ingredient after the given action. All the ingredients from list should be used, and their usage quantities match those given on the ingredient list. Finally, some recipes use references to a step number of prior actions, which makes the step dependent on other steps and their ordinal numbers.

We considered using the *Recipe1M+* dataset for our task, but it became clear that it has certain limitations regarding the validity of the recipe structure. To investigate these issues, we prepared a set of *corresponding recipes* built of $350,141$ pairs of recipes, identified by the same *URL*. This implies that both recipes in the pair, originated in the same place. They are considered a duplicate, despite not having exactly the same content. Example differences in content, as a result of different processing techniques is presented in Table 1. The set of *corresponding recipes* can be divided into two subsets - *Recipe1M+* subset (R_s) and *Gathered* subset (G_s).

During the data exploration process, we noticed that the number of instructions in the corresponding recipes varies, usually it is larger in R_s. To explain this difference, we manually verified 100 randomly selected pairs of corresponding recipes and found that 34 of them had the same structure, and in 62 cases recipes from R_s were malformed - had more steps than the original ones, while recipes from G_s kept the original structure.

We discovered that the recipe instructions in the *Recipe1M+* dataset might have been segmented into sentences instead of actual steps (see Table 1). To find out whether this explanation is correct, we split recipe instructions form G_s into sentences. The distribution of the number of obtained sentences in the recipe is similar to the R_s instructions distribution, which indicates that R_s recipes structure might have been altered. As our efforts aimed at generating semi-structured text, any changes in the structure of the documents are not acceptable.

Another issue we encountered during the data exploration, is the absence or malformation of fractions which we observed in *Recipe1M+* (see Table 1). We manually checked the same randomly selected 100 pairs and found, that 79 recipes from the R_s dataset missed at least one fraction from the set of ingredients, while the recipes from G_s were correctly reflecting the actual fractions in all cases. Furthermore, we found that the total number of recipes that had zero fractions was five times greater in R_s than in G_s.

Distortion of the fractions in this scale makes quantitative analyses pointless. Moreover, the text generator trained on this data would be unable to create logically coherent lists of ingredients.

4 *RecipeNLG* dataset

The results presented in Section 3 indicate the need for an enhanced dataset, appropriate for semi-structured text generation. We prepared a novel dataset named *RecipeNLG*, built on top of *Recipe1M+*, but enhanced with new and corrected records. Additional recipes were gathered from multiple cooking web pages, using automated scripts in a web scraping process.

4.1 Dataset cleansing

During the exploratory data analysis multiple problems regarding the structure of recipes were found and corrected. Recipes without any ingredients or instructions were considered to be extraction errors and were removed. We removed the excessive whitespace characters and replaced unicode symbols, (e.g., fractions) with their ASCII equivalents. Finally, the *Recipe1M+* dataset was appended to the gathered data. The *RecipeNLG* dataset contains an additional column, that identifies the origin of each record - *Recipe1M+* or *Gathered* data.

Deduplication was required to ensure that records do not overlap in the resulting set of the recipes. We began with finding duplicated recipes identified by the same *URL* - recipes downloaded from the same source are supposed to be identical. Then, pairs consisting of the same sequence of characters in instructions and ingredients were detected and removed. Finally, we found and removed near matches. The cosine similarity score was calculated pairwise upon a TF-IDF representation of the recipe ingredients and instructions.

Based on the *corresponding recipes* set (Section 3), we have determined the value of a *duplication threshold* as the minimum value of cosine similarity, starting from which a pair of records is considered to be a duplicate, by comparing the set of known duplicates with the set of candidate duplicates for each threshold value (Figure 2). For the *duplication threshold*, we chose the value where

Classic Chicken Tenderloin from www.food.com/recipe/classic-chicken-tenderloin-410132	
Recipe1M+	**RecipeNLG**
Ingredients missing slash character:	**Valid ingredients:**
• 1 lb chicken breast tenders • 12 cup Italian dressing • 1 teaspoon fresh lime juice • 1 12 teaspoons honey	• 1 lb chicken breast tenders • 1/2 cup Italian dressing • 1 teaspoon fresh lime juice • 1 1/2 teaspoons honey
Directions split into phrases:	**Valid directions split:**
• Drain and discard spices from the Italian dressing. • (Some may elect to keep the spices; the recipe will still turn out but will have a different flavor than intended. •). • Combine dressing, lime juice, and honey. • Marinate the chicken tenders in this mixture for at least one hour. • Grill chicken to a lightly golden color.	• Drain and discard spices from the Italian dressing. (Some may elect to keep the spices; the recipe will still turn out but will have a different flavor than intended.). • Combine dressing, lime juice, and honey. • Marinate the chicken tenders in this mixture for at least one hour. • Grill chicken to a lightly golden color.

Table 1: Comparison of two different representations of the same recipe

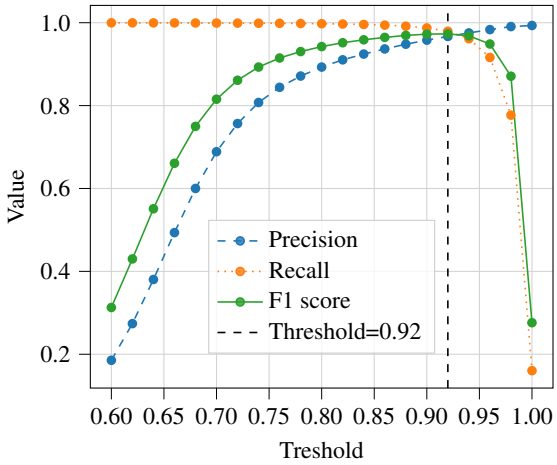

Figure 2: Cosine similarity threshold value selection for a dataset deduplication task.

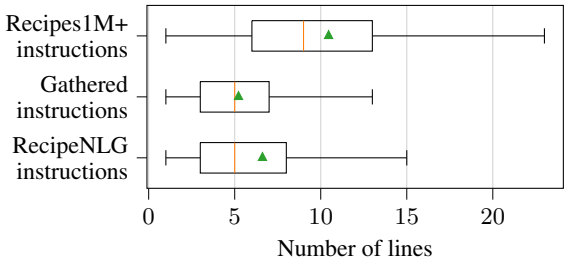

Figure 3: Comparison of number of lines of instructions between datasets. Triangles denote mean values.

the *F1 score* was the highest, which is 0.92. During the deduplication 523,040 records were removed.

We filtered out recipes in languages other than English. To recognize language of the recipe, we used only instructions, since foreign names (e.g., *croissant*) are common in titles and ingredients names, and may mislead the classifier. We used Google Translate API for language detection task.

4.2 *RecipeNLG* metrics

The *RecipeNLG* dataset contains $2,231,142$ distinct cooking recipes and to the best of our knowledge, it is the largest available dataset in the domain.

Figure 3 presents distributions of the number of elements in instructions, it visualizes the trend described in Section 3. This suggests, that recipes in *RecipeNLG* are more likely to have a structure consistent with the original recipes.

5 Experiment

We present our experiment performed on the *RecipeNLG* dataset. The goal was to prepare a model, which makes use of food entities to generate a complete cooking recipe. To accomplish this task, we prepared a NER model for identifying and extracting food entities. A GPT-2 model was fine-tuned for the recipe generation. The generated recipes were compared against the original recipes, using automatic evaluation metrics.

5.1 Identifying food entities

To use the NER for this problem, it was necessary to teach it what ingredients are. In order to determine the collection of ingredients, a subset of 500 recipes was manually annotated. This training data allowed us to extract food entities from the rest of the dataset. In total, the chosen recipes contained about 2,400 individual ingredients. We created the *penalty* metric to evaluate how precisely the model extracts a food entity (set of tokens $\hat{T}$) from an ingredient, based on a test set (set of tokens T).

$$penalty(\hat{T}, T) = \begin{cases} 0 & \text{if } \hat{T} = T \\ 0.5 & \text{if } \hat{T} \subset T \\ 1 & \text{if } \hat{T} \cap T = \emptyset \end{cases} \quad (1)$$

Since we allow partial matching of the result and the classified ingredient, we decided not to use standard metrics, such as precision and recall to evaluate NER performance.

5.2 Generating recipes from food entities

As a proof of concept for the usage of our dataset, we have created a language model based on the Hugging Face (Wolf et al., 2019) implementation of the pretrained GPT-2 (Radford et al., 2019).

Before training, we performed several post-processing operations on the dataset to ensure it is ready for our use case. It was crucial to create a model that generates "rich", extensive recipes. We decided to remove recipes with very short titles or instructions sections. We also removed recipes which contain phrases: 'step' in instructions, to remove the possibility of cross-step references based on ordinal numbers, and 'mix all', which lead the model to a preference of mixing everything over preparing detailed instructions.

The model was given a set of food entities and ordered to generate full recipes. A set of control tokens (visible on Figure 1) was prepared and embedded in the dataset. This has allowed the model to understand the recipe's underlying structure. Both the original recipes and the extracted food entities were used to prepare the training input. We placed multiple tokenized recipes into one context to speed up the training process. If the training sample was still shorter than the required size, the remaining space was filled with end of recipe tokens.

5.3 Evaluation

We selected a set of 100 recipes that were not used in training, to form a gold standard. Based on the food entities of each record from the gold standard 10 recipes were generated using two models: one trained on *RecipeNLG*, and one trained on *Recipe1M+* dataset. This resulted in 2000 generated recipes used to evaluate these two models.

Firstly, we used **cosine similarity** calculated upon TF-IDF representation to measure the similarity of a generated recipe and its gold standard counterpart. The results have shown that a *RecipeNLG* model generates recipes more similar to the gold standard than the *Recipe1M+* model (0.666 and 0.589 average cosine similarity, respectively).

We used the LanguageCheck spell and grammar checker to calculate the **amount of linguistic mistakes** - a metric that allowed us to estimate the overall performance of the model, and is applicable for a variety of texts. We calculated the average number of errors per recipe. There were fewer errors in the *RecipeNLG* model (2.78) than in the *Recipe1M+* (7.35). Interestingly, the *RecipeNLG* model scored better than the gold standard (3.64).

	BLEU	GLEU	WER
Recipe1M+	0.844	0.625	0.786
RecipeNLG	**0.866**	**0.662**	**0.751**

Table 2: Results of machine translation metrics for GPT-2 models fine-tuned on different datasets.

The last approach to the evaluation was the utilization of **translation metrics**. We used three common ones: BLEU (Papineni et al., 2002), GLEU (Wu et al., 2016), and WER (Word Error Rate). Scores achieved by each set are outlined in Table 2. The model trained on our dataset scored better on all of the translation metrics.

6 Conclusions & Future work

While the *RecipeNLG* dataset is based on the *Recipe1M+* dataset, it greatly expands the number of recipes available. What is even more important, the dataset comes with a changed scope – we didn't follow the idea of linking cooking recipes with their images, putting emphasis on the recipe text, structure and underlying logic. The new dataset provides over **1 million** new, preprocessed and deduplicated recipes on top of the *Recipe1M+* dataset. To the best of our knowledge, it is the largest publicly available dataset in the domain.

Our dataset, contrary to *Recipe1M+*, preserves unmodified ingredients quantities. It creates an opportunity to evaluate if the quantities are correctly generated by the model. In the future works, it could allow their normalization to a specific amount of servings. Another interesting potential work is on unification of mostly ambiguous units (e.g. cups, pinch) with regards to the item they are describing, which could have many uses in and outside of the culinary world, and further unification using knowledge graphs (Lawrynowicz, 2020).

The challenges we faced can be generalized to the other examples of text generation tasks. Therefore, we make this dataset public, expecting that it could enable new research in the area.

Acknowledgments

Model prototyping was supported with Cloud TPUs from Google's TensorFlow Research Cloud (TFRC) programme.

References

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – mining discriminative components with random forests. In *Computer Vision – ECCV 2014*, pages 446–461, Cham. Springer International Publishing.

P. Buneman. 1997. Semistructured data. In *PODS '97*.

Khyathi Chandu, Eric Nyberg, and Alan W Black. 2019. Storyboarding of recipes: Grounded contextual generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

M. F. K. Fisher. 1969. *The Anatomy of a Recipe, With Bold Knife and Fork*. Counterpoint.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339, Austin, Texas. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Agnieszka Lawrynowicz. 2020. Creative AI: A new avenue for the Semantic Web? *Semantic Web*, 11(1):69–78.

Helena H. Lee, Ke Shu, Palakorn Achananuparp, Philips Kokoh Prasetyo, Yue Liu, Ee-Peng Lim, and Lav R. Varshney. 2020. RecipeGPT: Generative pre-training based cooking recipe generation and evaluation system. *In Companion Proceedings of the Web Conference 2020*.

Ming Liang and Xiaolin Hu. 2015. Recurrent convolutional neural network for object recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Weiqing Min Luis Herranz and Shuqiang Jiang. 2018. Food recognition and recipe analysis: integrating visual content, context and external knowledge. *ArXiv*, abs/1801.07239v1.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In *EMNLP*, pages 5975–5981.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *ArXiv preprint ArXiv:1708.02182*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Md Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2373–2383. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Amaia Salvador, Michal Drozdzal, Xavier Giró, and Adriana Romero. 2019. Inverse cooking: Recipe generation from food images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10445–10454.

Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marín, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning cross-modal embeddings for cooking recipes and food images. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3068–3076.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith

Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP (2018)*, pages 1358–1368.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 1850–1859.